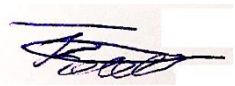
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Факультет електроніки та інформаційних технологій
Кафедра електроніки, загальної та прикладної фізики

Кваліфікаційна робота магістра

IoT системи на основі мікроконтролерів ESP: концепція мережі та програмне забезпечення

Студент групи ЕП.м – 91н  В.Г. Кучменко

Науковий керівник,
к. ф.-м. н., ст.викладач  К. В. Тищенко

Завідувач кафедри електроніки,
загальної та прикладної фізики,
д.ф.-м.н., професор  І.Ю.Проценко

Суми – 2021

РЕФЕРАТ

Об’єктом дослідження в науковій роботі є побудова IoT систем об'єднання об'єктів в мережу для покращення їхньої функціональності фізичних величин на основі мікроконтролерів сімейства ESP.

Метою роботи є вивчення особливостей та технічних характеристик мікроконтролерів ESP а також реалізація IoT систем і мереж віддаленого керування фізичним експериментом з їх використанням.

Під час виконання роботи використовувалась платформа NodeMCU на базі мікроконтролера ESP-8266

У результаті проведених досліджень показано, що система IoT фізичних величин та керування експериментом може бути побудована на базі мікроконтролерів сімейства ESP. Широко розповсюджені готові плати збору даних, наприклад NodeMCU або ESPduino на контролерах ESP дозволяють швидко розробити прилад для керування фізичним експериментом, вони підтримують велику кількість плат розширення та мають низьку вартість.

Великою перевагою контролерів ESP є присутність на платі вбудованого контролера безпровідного зв’язку, що дозволяє використовувати його підключивши до мережі WiFi і реалізовувати передачу даних із використанням стандартних мережевих протоколів.

Робота викладена на 41 сторінці, зокрема містить 11 рисунків, список цитованої літератури із 21 джерела.

КЛЮЧОВІ СЛОВА: КОНТРОЛЕР, МІКРОКОНТРОЛЕР, ESP8266, IoT, ESP, ЗБІР ДАНИХ, КЕРУВАННЯ.

ЗМІСТ

ВСТУП

В наш час використання складних систем спостереження, для керуванні фізичним експериментом, дослідженням, контроль є запорукою успішної, безпечної і економічно вигідної роботи експериментального обладнання. Базовими пристроями систем такого типу є датчики різноманітних систем та величин, данні які отримується і аналізується за допомогою систем збору даних. Однією із таких систем є сімейство мікроконтролерів ESP, які будуть розглядатися в даній роботі.

Основною задачею є автоматизація процесів, що дозволяє економити ресурси та час, не тільки у промисловості, але й в житті людини. На даний час існує велика кількість апаратних, та програмних рішень, котрі покликані на покращення ефективності розробки систем керування. Актуальним також є питання створення такої системи власноруч на основі модулів, що виготовляються серійно.

До плат на основі контролерів ESP можна підключати різні периферійні пристрої – датчики фізичних величин, кнопки, світлодіоди, різноманітні пристрої виведення. Написавши програму, можна забезпечити функціонування всіх підключених пристроїв за певними алгоритмами, перетворюючи розподілену систему в один закінчений електронний прилад.

Важливим є питання інтеграції вимірювальної системи, що є необхідним атрибутом системи керування. Така можливість забезпечується вбудованим дисплей модулем, здатним працювати у всіх популярних режимах. Також підключаючи до мікроконтролера різні модулі – Shields, можна розширювати базовий функціонал платформи.

Мета роботи – розробка та виготовлення IoT системи з використанням мікроконтролерів ESP.

Результати роботи представлені і обговорені на Науково-технічній конференції «Фізика, електроніка, електротехніка, ФЕЕ-2021».

РОЗДІЛ 1
МІКРОКОНТРОЛЕРНІ СИСТЕМИ В МЕРЕЖАХ IoT

Мікроконтролер – це спеціальний чіп, який використовується для управління різними електронними пристроями. Мікроконтролери розроблялись паралельно із мікропроцесорами загального призначення, і великою мірою застосовують однакові архітектурні підходи до розробки та програмування.

У розробників мікроконтролерів була революційна ідея: поєднати процесор, пам’ять, ПЗУ та периферію в одному блоці, подібному до звичайного чіпа. З тих пір щорічне виробництво мікроконтролерів часто перевищує виробництво процесорів, і їх попит не зменшується [1].

Мікроконтролери виробляються декількома компаніями, причому не тільки сучасними 32-розрядними мікроконтролерами, але також 16 і навіть 8-розрядними мікроконтролерами (i8051 та аналоговими). У кожній сімействі ви часто знайдете майже однакові моделі, які відрізняються швидкістю процесора та ємністю пам'яті.

Особливістю мікроконтролерів є те, що вони в основному використовуються в бортових системах, іграшках, машинах та домашній автоматизації, де не потрібна потужність центрального процесора, але достатній баланс між ціною та функціональністю.

З цієї причини старі типи мікроконтролерів все ще дуже поширені і пропонують безмежну кількість можливостей: від автоматичного відкривання дверей до поливу газону, включаючи інтеграцію в систему «Розумний дім». У той же час існують більш потужні мікроконтролери, які можуть виконувати сотні мільйонів операцій в секунду для підключення великої кількості пристроїв. Враховуючи мінливість параметрів і функцій, розробник спочатку оцінює завдання, а потім відбирає найбільш підходящий матеріал для його реалізації [2].

На сьогоднішній день існує понад 200 модифікацій мікроконтролерів, сумісних з i8051, виготовлених двома десятками компаній та безліччю інших типів мікроконтролерів. 8-розрядні мікроконтролери PIC від Microchip

Technology та AVR від Atmel, мікроконтролери TI MSP430 та 32-розрядні, архітектура ARM, розроблена ARM Limited, та необхідні ліцензії на їх виробництво будуть продані іншим компаніям.

Мікроконтролер характеризується великою кількістю параметрів, оскільки являє собою складний пристрій, керований пристроєм та електронним пристроєм (мікросхемою). Префікс "мікро" у назві мікроконтролера означає, що він виготовлений в мікроелектроніці.

Під час роботи мікроконтролер зчитує команди з пам'яті або вхідного порту і виконує їх. Значення кожної команди визначається системою управління мікроконтролером. Система команд вбудована в архітектуру мікроконтролера, і виконання коду інструкцій виражається у внутрішніх елементах мікросхеми певних мікрооперацій.

За допомогою мікроконтролерів ви можете чудово керувати різними електронними та електричними пристроями. Деякі моделі мікроконтролерів настільки потужні, що можуть безпосередньо перемикати реле (наприклад, у різдвяних вінках). Як правило, мікроконтролер працює не поодинці, а вбудований в схему, в яку також підключаються дисплеї, входи клавіатури, різні датчики тощо. [3]

Програмне забезпечення мікроконтролера може привернути увагу кожного, хто хоче максимізувати оптимізацію програмного коду, оскільки пам'ять мікроконтролерів зазвичай становить від 2 до 128 КБ. Якщо він менший, вам слід написати на Assembler або Forty, якщо це можливо, а потім скористатися спеціальними версіями BASIC, Pascal, але переважна більшість мікроконтролерів запрограмовані на C++. Перш ніж нарешті програмувати мікроконтролер, він тестується на емуляторах: програмному чи апаратному [1].

З огляду на вищевикладене, ставиться питання про різні властивості мікросхем та мікроконтролерів та особливості їх застосування для реалізації проектів автоматизації фізичного досвіду та їх використання в промисловій електроніці.

1.1 Структура мікроконтролера

Розглянемо основні структурні компоненти мікроконтролера на прикладі мікросхем із сімейства AVR (рис. 1.1). Мікроконтролер AVR містить: швидкий процесор RISC, два типи незалежної пам’яті (пам’ять Flash-програми та пам’ять даних EEPROM), оперативну пам’ять, порти вводу-виводу та різні схеми периферійного інтерфейсу [4].

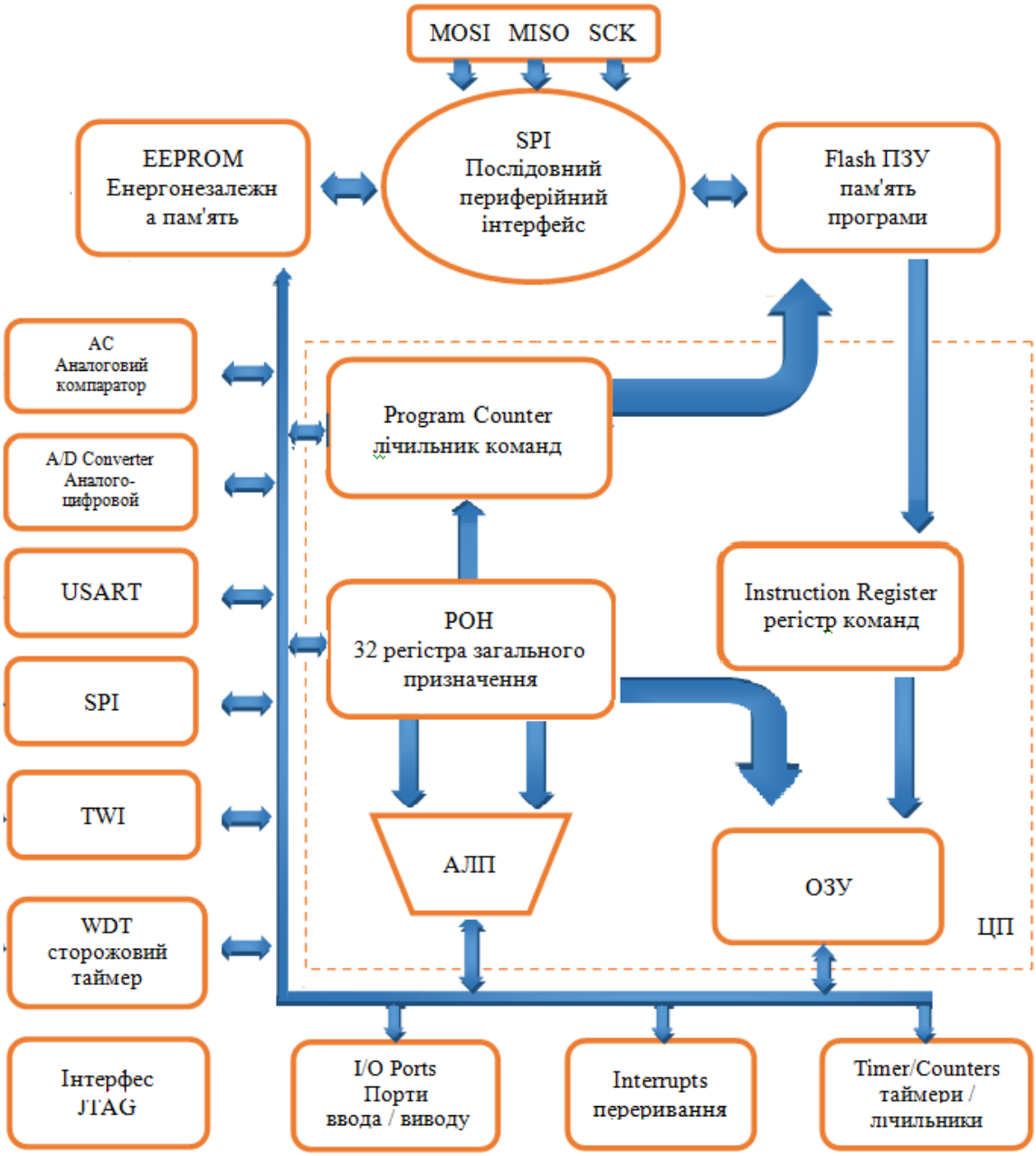


Рис. 1.1. Структура мікроконтролера. Адаптовано з роботи [1].

1.2 Процесор

Основою мікроконтролерів AVR є 8-бітове мікропроцесорне ядро або центральний процесор (CPU), заснований на принципах архітектури RISC (Reduced Instruction Set Computing). Процесор із зменценим набором команд. Система управління має спрощений вигляд. Усі команди мають однаковий формат із простим кодуванням. Доступ до пам'яті можна отримати за допомогою команд завантаження та збереження. Усі інші команди чутливі до регістру. Команда, яка надходить до центрального процесора, вже розділена на поля і не вимагає додаткового дешифрування. Оскільки інструкції RISC є простими, вони вимагають менше логіки для виконання, що в кінцевому рахунку зменшує витрати на процесор. Однак більшість сьогоднішнього програмного забезпечення було написано та скомпільовано спеціально для процесорів Intel CISC. Для використання архітектури RISC поточні програми повинні бути перекомпільовані, а іноді і переписані.

Основою цього блоку є арифметично-логічний пристрій (АЦП). Відповідно до сигналу системного тактового сигналу з пам'яті програми відповідно до вмісту лічильника команд (лічильник програм - ПК) вибирається наступна команда і виконується АЦП. При виборі команди з пам'яті програми виконується попередньо вибрана команда, і може бути досягнута швидкість від 1 MIPS до 1 МГц [4].

АЦП підключений до регістрів загального призначення (GPR). Є лише 32 регістри загального призначення, і вони мають байтовий формат, кожен з яких складається з восьми бітів. RZP розташовані на початку адресного простору оперативної пам'яті, але фізично не є його частиною. Тому доступ до них можна отримати двома способами (регістри та пам’ять). Це рішення є особливістю AVR [5] і підвищує ефективність та продуктивність мікроконтролера.

Різниця між регістрами та оперативною пам’яттю полягає в тому, що будь-яка операція може бути виконана над регістрами (арифметичними, логічними, бітовими), а в оперативну пам’ять можна записувати лише дані з регістрів.

1.3 Пам’ять

Мікроконтролери AVR реалізують "Гарвардську" архітектуру, відповідно до того, що на місцях розподілені простори управління пам'яті програми та пам'яті даних, а також шину доступу до них. Кожна з областей пам'яті даних (RAM та EEPROM) також буде знаходится в своему адресному просторі.

Пам'ять програми розрахована на послідовність команд, що контролюють функціонування мікроконтролера, і має організацію з 16-ти біт. Усі AVR керуються програмами флеш-пам'яті, які ви можете використовувати в різних розмірах, від 1 до 256 КБ. Його головна перевага полягає в тому, що він заснований на принципі електричного перепрограмування, тобто розшифровує, дозволяє розмити множинні та записи інформації. Програма запускається у флеш-пам'яті AVR як за допомогою звичного програматора, так і за допомогою інтерфейсу SPI, включеного безпосередньо в зібрану плату [6].

Усі мікроконтролери сімейства Mega мають можливість програмувати себе, навіть самостійно змінювати незалежно від пам'яті з своєї програми. Ця функція дозволяє створювати на основі дуже гнучких систем, алгоритм яких буде змінювати сам мікроконтролер залежно від внутрішніх умов чи зовнішніх подій [6].

Кількість гарантованих циклів перезапису флеш-пам'яті на мікроконтролерах AVR другого покоління не менше 10000 циклів має типове значення 100000 циклів. (Офіційна технічна документація Atmel Corp. вказує вартість 10000 циклів)

Пам'ять даних поділяється на 3 частини: регістрова пам'ять, оперативна пам'ять (RAM - оперативна пам'ять RAM) і незалежна пам'ять (EEPROM або EEPROM).

Реєстрова пам'ять включає в себе 32 записи загального призначення (РЗП або GPR), об’єднані в один файл та записи служб вводу-виводу (РВВ). Обидві вони розташовані в адресному просторі оперативної пам'яті, але вони не є його частиною.

В області реєстраторів вводу-виводу існують різні сервісні регістри

(регістри управління мікроконтролером, регістри стану тощо), як і регістри управління периферійними пристроями, які є частиною мікроконтролера. По суті, управління мікроконтролером полягає в управлінні цими записами. [7]

Рис. 1.2. Структура пам’яті мікроконтролера. Адаптовано з роботи [7].

Пам'ять EEPROM використовується для тривалого зберігання різної інформації, яка може змінюватися під час роботи системи мікроконтролера. Усі AVR мають 64-байтний, 4 Кб, електрично енергонезалежний, перезаписуваний блок зберігання даних EEPROM. Цей тип пам'яті, який безпосередньо доступний програмі мікроконтролера під час її роботи, підходить для зберігання різних проміжних даних. константи, коефіцієнти, серійні номери, ключі тощо. зовнішньо завантажений програміст. Кількість циклів стирання / запису: принаймні 100 000

Внутрішня статична оперативна пам'ять (SRAM) має байтовий формат і використовується для онлайн-зберігання.

Розмір основної пам'яті може змінюватися на різних мікросхемах від 64 байт до 4 КБ. Кількість циклів читання та запису в оперативній пам'яті не обмежена, але якщо напруга живлення буде знижена, вся інформація буде втрачена. За допомогою деяких мікроконтролерів можна налаштувати зовнішнє статичне підключення до оперативної пам'яті розміром до 64 КБ [1].

1.4 Периферійні пристрої

Сімейство мікроконтролерів AVR включає всі запам'ятовуючі пристрої, крім пам'яті (ОЗП та ПЗП) та процесора, а саме: порти (від 3 до 48 ліній вводу-виводу), підтримка зовнішніх переривань, лічильники таймерів, сторожовий таймер, аналогові компаратори, 10 розрядний на 8 канальний АЦП , Інтерфейси каналів UART, JTAG та SPI, пристрої скидання для зменшення потужності, модулятори ширини імпульсу. Пристрої також містять канали вводу-виводу (цифрові та аналогові).

Порти вводу-виводу AVR мають кілька незалежних ліній вводу / виводу від 3 до 53. Кожну лінію підключення можна запрограмувати на вхід або вихід. Потужні вихідні драйвери пропонують номінальний струм 20 мА на лінію підключення (вхідний струм) до максимального значення 40 мА і дозволяють, наприклад, безпосереднє підключення до світлодіодів мікроконтролера та біполярних транзисторів. Сумарне струмове навантаження всіх ліній у з'єднанні не повинно перевищувати 80 мА (усі значення стосуються напруги живлення 5 В) [8].

Архітектурною особливістю портів вводу-виводу AVR є те, що для кожного фізичного входу передбачено 3 біти управління або контролю замість 2, як у традиційних 8-бітних мікроконтролерах (Intel, Microchip, Motorola тощо). Це дає можливість уникнути необхідності мати копію вмісту порту в пам'яті з міркувань безпеки та збільшує швидкість роботи мікроконтролера під час роботи із зовнішніми пристроями, особливо в умовах зовнішніх електростатичних перешкод.

Система переривань є однією з найважливіших частин мікроконтролера. Всі мікроконтролери AVR мають багаторівневу систему переривання. Вони переривають звичайний потік програми для виконання пріоритетного завдання через внутрішню або зовнішню подію.

Для кожної події розробляється окрема програма, яка називається підпрограмою обробки запиту на переривання, і зберігається в пам'яті програми.

Коли відбувається подія переривання, мікроконтролер зберігає вміст лічильника команд, перериває виконання поточної програми центральним процесором і виконує програму обробки переривань.

Після виконання процедури переривання попередньо збережений лічильник команд скидається, і процесор повертається до виконання перерваної програми [9].

Ви можете встановити пріоритет для кожної події. Термін пріоритет означає, що виконується процедура переривання і може бути перервана іншою подією, лише якщо вона має вищий пріоритет, ніж поточний. В іншому випадку ЦП не буде обробляти нову подію, поки не буде оброблена попередня подія.

Мікроконтролери AVR складаються з 1 до 4 таймерів / лічильників 8 або 16 бітів, які можуть функціонувати як внутрішні таймери джерела тактових сигналів та зовнішні лічильники подій (переривань) [10].

Вони можуть використовуватися для створення точних інтервалів часу, підрахунку імпульсів на штифтах мікроконтролера, підготовки імпульсного курсу та синхронізації приймача з послідовним каналом зв'язку. У режимі ШІМ таймер / лічильник може бути модулятором ширини імпульсу і використовується для генерації сигналу на програмованій частоті. Таймери / лічильники можуть обробляти запити на переривання, модифікуючи процесор для обробки подій і позбавляючи його від необхідності періодичного опитування стану таймерів. Оскільки мікроконтролери в основному використовуються в системах реального часу, таймери / лічильники є одним з найважливіших елементів [11].

Таймер WatchDog призначений для уникнення катастрофічних наслідків випадкових помилок програми. Він має власний RC-генератор, який працює на частоті 1 МГц. Як і внутрішній основний RC-генератор, значення 1 МГц є приблизним і залежить головним чином від розміру напруги та температури мікроконтролера.

Ідея використання таймера спостереження надзвичайно проста і полягає в тому, щоб періодично перезапускати його під контролем програми або зовнішнього впливу, доки налаштування процесора не будуть скинуті. Якщо

програма працює нормально, команду скидання таймера команди потрібно виконувати періодично, щоб уникнути перезапуску процесора. Якщо мікропроцесор випадково виходить із програми (наприклад, через сильні перешкоди в ланцюзі живлення) або повторюється в частині програми, команда вимкнення таймера скидання навряд чи буде успішною. тривалий час і поверне процесор до роботи.

Аналоговий компаратор порівнює напруги на двох виходах мікроконтролера. Результатом порівняння є логічне значення, яке можна прочитати в програмі.

Аналого-цифровий перетворювач дозволяє отримати цифрове значення напруги, що подається на його вхід. Цей результат тимчасово зберігається в реєстрі даних аналогово-цифрового перетворювача. Який із контактів мікроконтролера буде входом АЦП, визначається числом, введеним у відповідний регістр [11].

Універсальний синхронний / асинхронний передавач і приймач (UART) – представляє собою послідовний інтерфейс для організації інформаційного каналу для обміну мікроконтролерами із зовнішніми пристроями. Він може працювати в дуплексному режимі (одночасно надсилаючи та отримуючи дані). Він підтримує стандартний комунікаційний протокол RS-232, котрий дозволяє спілкуватися з персональним комп'ютером.

Послідовний периферійний інтерфейс SPI (Peripheral Serial Interface) використовується для обміну даними між двома пристроями. Він може використовуватися для передавання пакетів даними між мікроконтролером та різноманітними зовнішніми пристроями, наприклад цифровими потенціометрами, аналогово-цифровими та цифро-аналоговими перетворювачами, постійними запам’ятовуючими пристроями на основі флеш пам’яті та іншим периферійним обладнанням. Використовуючи можливості цього інтерфейсу можливе забезпечення зручного обміну даними між декількома мікроконтролерами з підтримкою протоколу SPI. Крім того, мікроконтролер можна програмувати через інтерфейс SPI.

Двопровідний послідовний інтерфейс (TWI) повністю аналогічний базовій версії інтерфейсу Philips I2C, і його можна застосовувати для підключення різних пристроїв (до 127) посередництвом двобітового комунікаційного каналу, котрий складається з тактової лінії (SCL) та лінії передачі даних (SDA).

Інтерфейс JTAG був розроблений командою експертів з тестування електронних компонентів (Joint Test Action Group) і зареєстрований як галузевий стандарт IEEE Std 1149.1-1990. Чотири інтерфейси JTAG використовуються для тестування друкованих плат, налагодження схем та програмування мікроконтролерів [12].

Багато мікроконтролерів сімейства Mega мають інтерфейс JTAG або debugWIRE, сумісний з IEEE Std 1149.1, для інтегрованої налагодження. Крім того, усі Mega мікроконтролери з флеш-пам'яттю 16 КБ або більше можна програмувати через JTAG інтерфейс.

Тактовий генератор генерує імпульси для синхронізації роботи всіх вузлів мікроконтролера. Внутрішній тактовий сигнал AVR може запускатися за допомогою декількох опорних джерел частоти (зовнішній генератор, зовнішній кристалічний резонатор, внутрішній або зовнішній RC-ланцюг). Мінімально допустима частота не обмежена. Гранична робоча частота визначається конкретним типом мікроконтролера та описана специфікацією компанії розробника (Atmel) у його технічних характеристиках, хоча майже всі мікроконтролери AVR із заданою робочою частотою, наприклад 10 МГц при кімнатній температурі [8], можуть бути прискорені до роботи на частоті 12 МГц або більше.

RTC реалізований у всіх Mega контролерах та у двох "класичних" кристалах: AT90 (L) S8535. Таймер / лічильник RTC може бути підключений до основного джерела тактової частоти або додаткового асинхронного джерела опорної частоти (кристалічний резонатор або зовнішній тактовий сигнал) під управлінням програмою. Для цього на мікросхемі зарезервовано два висновки. Внутрішній генератор оптимізований для роботи із зовнішнім кристалічним резонатором 32,768 кГц [12].

1.5 Мікроконтролер ESP8266

Кожен, хто використовує Arduino у своїх проектах, уже створив власний портрет сімейства плат ESP і представив його як просту карту розширення для бездротового підключення до Інтернету. Насправді це більше мікроконтролер з декількома своїми інтерфейсами.

Модуль ESP8266 (рис. 1.2) має 11 універсальних I/O портів, включаючи 9 цифрових з підтримкою широтно-імпульсної модуляції (ШІМ), аналоговий вихід для аналого-цифрового перетворювача (АЦП), підтримку UART та SPI для розробки повністю автономних команд. Саме всі ці переваги чіпа зробили його дещо революційним під час його розробки. Якщо ви розробляєте системи та робототехніку на базі Arduino або РПІ , вам слід використовувати ESP8266 та протестувати його на своїх проектах. Короткий час відгуку, високошвидкісні інтерфейси та можливість використання декількох протоколів одночасно в багатоканальному режимі відкривають безліч можливостей для реалізації автономних автоматизованих систем з передачею даних за бездротовими каналами зв'язку. [12]

Мікроконтролер вже є у великій кількості напівпрофесійних пристроїв, оскільки його більш дорогі конкуренти не можуть йому все ж нічого протиставити.. Однак, якщо у вас не бюджет у десятки тисяч доларів, вам навряд чи знадобиться щось краще. На карті користувачів по всьому світу вже є десятки альтернативних програм, оскільки ви можете встановити їх на контролер за лічені секунди, щоб розширити основні функції, не купуючи додаткові адаптери та розхширення. В даний час мікроконтролер використовується у двох напрямках: з іншими мікроконтролерами, для яких програмне забезпечення пишеться окремо на Arduino, і з прямим підключенням до ПК та наступним управлінням через модуль UART.

Для інженерів та програмістів контролер є просто знахідкою, оскільки прошивка написана мовою високого рівня з великою кількістю інтерпретаторів. Мова Arduino універсальна та підходить для реалізації більшості алгоритмів.

Щоб оновити мікропрограму, просто підключіть пристрій через USB-модуль та встановіть нову прошивку за допомогою стандартної програми.

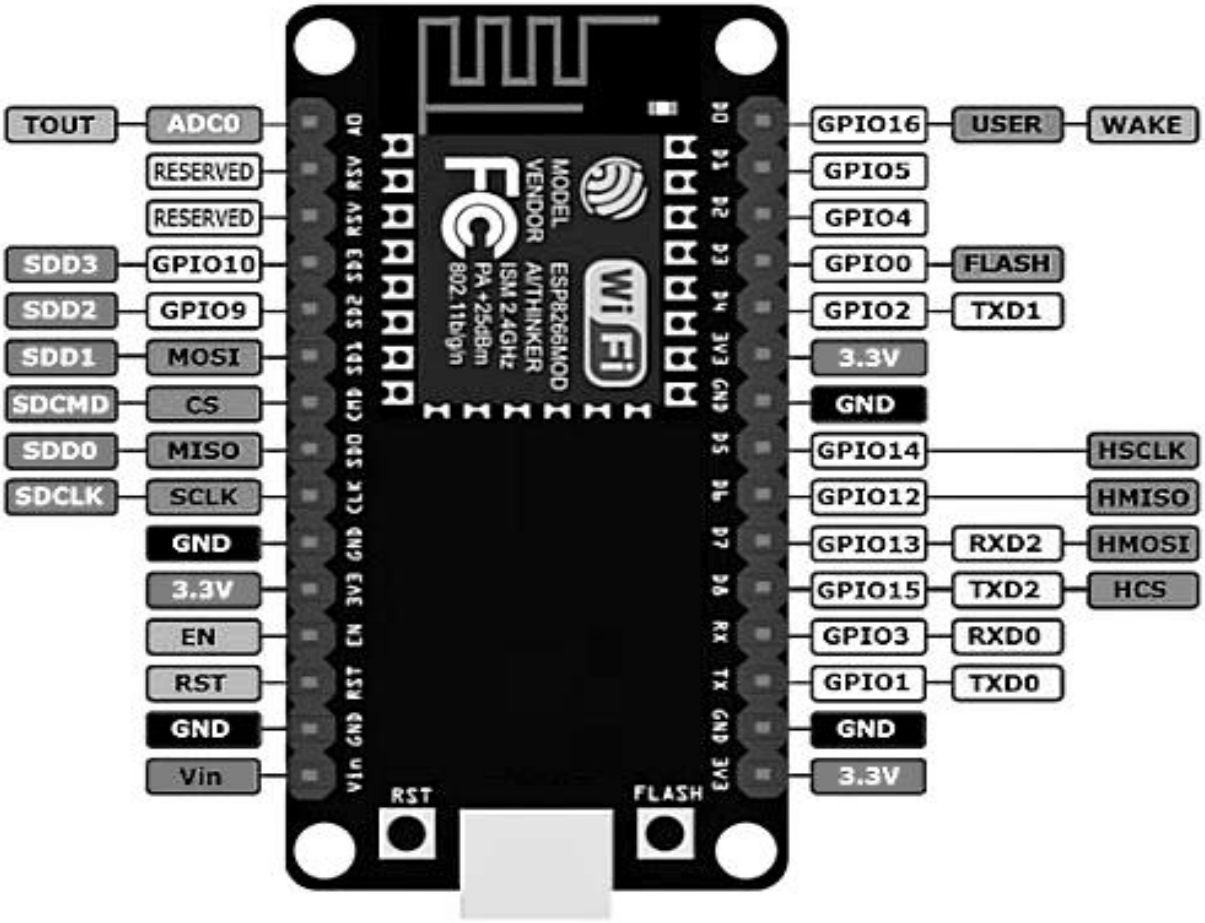


Рис. 1.2. Призначення портів модуля NodeMCU на базі контролера ESP8266 ESP–12E. Адаптовано з роботи [12].

Той самий перетворювач USB у TTL, який можна знайти на більшості апаратних пристроїв ПК, з можливістю підключення за допомогою додаткових плат і контролерів Arduino, робить пристрій надзвичайно універсальним.

1.6 Сфери застосування ESP8266

У попередніх розділах було показано, що контролер ESP8266 надзвичайно функціональний та гнучкий і може використовуватися в будь-якій системі, де вам потрібен контроль підключення до Інтернету. Іншими словами, якщо вам

потрібно лише отримувати та надсилати певні пакети за допомогою певних протоколів, щоб фільтрувати систему та захищати її від хакерів та DDOS-атак, пристрій буде корисним.

Можливість налаштування кількості пакетів, які отримує користувач за одиницю часу, робить це кращим, ніж традиційні карти розширення, що дозволяють системам отримувати доступ до Інтернет-з'єднання. Користувачі можуть писати власну програму для класифікації, надсилання та фільтрування пакетів без необхідності інтегрованих авторів та неоптимізованих алгоритмів без використання ООП. Однак розробники завжди дбали про тих, хто не вміє програмувати, і відразу ж представили на своєму веб-сайті список відповідних програм, які можна встановити в мікроконтролер в різних ситуаціях.

По суті, пристрій має широкий спектр використання, від можливості підключення найпростішого модему Wi-Fi до складних систем [13], таких як розумний дім. Нарешті, сам пристрій може розподілити захоплену мережу і таким чином стати точкою доступу, через яку ви також можете використовувати його як підсилювач сигналу і одночасно як фільтр з'єднання. Швидкість залишає бажати кращого, але все одно можна купити підсилювач.

Dupont з’єднувачі відповідають обраній платі. За бажанням ви можете використовувати кабель USB-TTL та інші кабелі відповідно до описаних вище протоколів підключення до ПК, а також перетворювач PL2303-FTDI. Ви повинні підключити один кінець кабелю до порту комп'ютера, і чіп повинен бути негайно розпізнаний. Потім просто завантажте відповідну утиліту для написання скриптів Arduino [13] або скористайтеся наявною прошивкою.

1.7 Класифікація МК архітектури обчислювальної системи

Перший МК назвав стандартну архітектуру CISC, яка застосовується в настільних комп'ютерах у той час. Особливості CISC: команди виконують почергово іншого друга та мають роздільну тривалість та структуру. Вибірка команд з пам'яті здійснюється побічно і виконується за кілька тактів. CISC-

архітектура надає: МК з сімейства Motorola HC05 / HC08, МК з ядром MCS-51, МК з сімейства Infineon C500 та ряд інших.

На початку 1980-х років була розроблена нова архітектура з багато об’єктивною назвою RISC (абревіатуру, запропоновану Д. Паттерсоном з Каліфорнійського університету в м. Берклі, США). Основна ідея в заміні складних командах однотипними простими та виконанням їх єдиним потоком на паралельному конвеєрі. Усі команди мають фіксовану тривалість і в ідеалі повинні виконуватись за один, а не за кілька, тактів, що досягається підвищеним швидким дією.

Одним з перших МК з архітектурою RISC став PIC-контролер 16C54 фірми Microchip. Завдяки високій продуктивності та тривалості десяткам легко заповіданих команд, PIC-контролери швидко поширюють популярність у всьому світі. Їх застосували після того, як розробники фірми Atmel, Scenix та ін.

RISC-архітектура в МК зараз є конкуренцією. Навіть новіші клони контролерів, що мають сумісність з MCS-51, відрізняються від прабатьків у першій черзі зміною архітектури. Це чітко простежується на прикладі продукції фірми Atmel - «стара» мікросхема AT89C2051 (CISC) проти покращених «нових» мікросхем AT89S2051, AT89LP2052 (RISC).

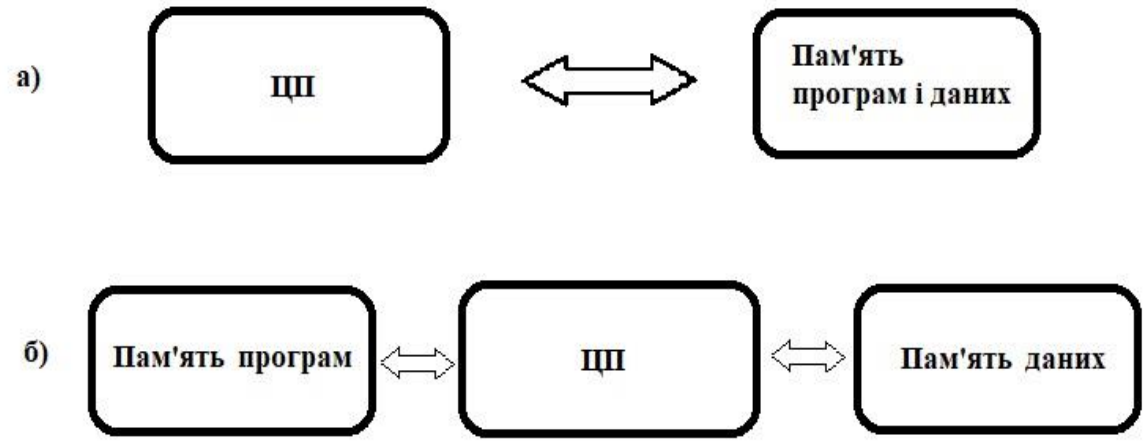


Рис. 1.3 Архітектура МК: а) Принстонська б) Гарвардська . Адаптовано з роботи [13].

З точки зору принципів конструювання вичислювальної системи виділяють принстонську та гарвардську архітектури. Обидві назви пов'язані з університетами в США.

Принстонська архітектура була розроблена Джоном фон Нейманом і незалежністю від академіка С.А.Лебедевого. Вона використовує загальну пам'ять для зберігання програм та даних. Основні переваги має в розповсюдження схемних технологій та в гнучкості розподілу ресурсів між областями пам'яті.

Особливістю гарварської архітектури є наявність розділених адресних просторів для зберігання команд та даних. Ця архітектура майже не використовувалася до кінця 1970-х років, розробивши МК накопичувачі, саме вона дала їм визначені переваги. Зокрема, аналіз реальних програм показує, що об'єм пам'яті даних МК використовується для зберігання змінних результатів, наприклад, для упорядкування менше необхідного обсягу програм пам'яті. Значить, можна скоротити розрядність шин даних, зменшити число транзисторів у мікросхемі, а також заохотити та прискорити доступ до інформації про зразки в обох «напівшарах» пам'яті. Як слідство, зараз більшість сучасних МК використовують RISC-архітектуру гарвардського типу.

РОЗДІЛ 2
ПОНЯТТЯ ТА ПРИНЦИП РОБОТИ КОМУНІКАЦІЙНОЇ МЕРЕЖІ INTERNET OF THINGS

IoT – інформаційних комунікаційних технологій (ІКТ), як очікується, стане революцією в передачі інформації від людини до людини, від людини до речей і від речей до речей. Розумні пристрої можуть підключатися, передавати інформацію та приймати рішення від імені людей. Ця нова технологія називається "підключення для будь-чого". Вони можуть підключатися де завгодно, у будь-який час і будь-що. Середовище IoT складається з величезної кількості розумних пристроїв, але з багатьма обмеженнями. Це об'єм пам’яті, короткий час живлення та радіодіапазон, є деякими з цих обмежень. Тому реалізація IoT вимагає протоколів зв'язку, які можуть ефективно керувати цими умовами [14].

У цьому розділі також буде розглянуто та порівняно протокол зв'язку IoT, який реалізується як чітке розуміння протоколу зв’язку IoT, їх плюси та мінуси, а також їх швидкість, живлення та робочий діапазон

Рис. 2.1 Протоколи зв'язку IoT. Адаптовано з роботи [14].

2.1. Широка територіальна мережа низької потужності (LPWAN)

SigFox – це технологія низької потужності для бездротового зв’язку різноманітного діапазону низько енергетичних об'єктів, таких як датчики та програми M2M. Він дозволяє передавати невеликі обсяги даних у межах до 50 кілометрів. SigFox використовує технологію Ultra Narrow Band (UNB). Ця технологія призначена лише для обробки низьких швидкостей передачі даних від 10 до 1 000 000 біт на секунду і може працювати на невеликому акумуляторі. Технологія NFC використовується в розумних лічильниках, моніторах користувачів, сільському господарстві, пристроях безпеки, вуличному освітленні та екологічних датчиках. Підтримка SigFox запускає мережеву топологію.

Cellular – це ідеально підходить для програм, які потребують високої пропускної здатності та мають джерело живлення додатку IoT, який вимагає роботи на більшій відстані. Він може скористатися можливостями стільникового зв'язку GSM / 3G / 4G, він може забезпечити надійне високошвидкісне підключення до Інтернету. Однак воно потребує високого енергоспоживання. Тому він не підходить для зв'язку M2M або локальної мережі. Протокол стільникового зв'язку також використовується для багатьох програм, особливо для додатків, що включають мобільні пристрої. [14].

Wi-Fi – це сімейство протоколів бездротової мережі, засноване на сімействі стандартів IEEE 802.11, Wi-Fi використовує декілька частин протоколів IEEE 802 і призначений для безперебійної роботи з дротовим Ethernet. Різні версії Wi-Fi визначаються різними стандартами протоколу IEEE 802.11, причому різні радіотехнології визначають діапазони радіозв'язку, а також максимальні діапазони та швидкості, яких можна досягти. Wi-Fi найчастіше використовує діапазони 2,4 гігагерц (120 мм) і 5 гігагерц (60 мм); ці смуги поділяються на кілька каналів. Канали можуть бути спільними між мережами, але тільки один передавач може локально передавати на каналі в будь-який момент часу.

Смуги хвиль Wi-Fi мають відносно високе поглинання та найкраще працюють для прямого огляду. Багато загальних перешкод, таких як стіни, стовпи, побутова техніка тощо, можуть значно зменшити дальність дії, але це також допомагає мінімізувати перешкоди між різними мережами в людних умовах.

2.2 Мережа короткого діапазону

6LoWPAN - це перший і найчастіше застосовуваний стандарт у протоколах зв’язку IoT, оскільки це стандартний протокол роботи в мережі Інтернет на основі IP. Він може бути з'єднаний безпосередньо з іншою мережею IP без проміжних об'єктів, таких як шлюзи або проксі сервера. Цей стандарт був створений інженерною робочою групою інтернету (IETF), стандартним зв'язком з інтернет-протоколом (IP) з низькою потужністю бездротові мережі IEEE802.15.4, що використовують IPv6.

Рис. 2.2 Приклади мереж. Адаптовано з роботи [14].

Він підтримує 2^{128} IP-адрес, тому кількість адрес більш ніж достатня. Це спрямовано на підтримку різної довжини адрес. Це також низька вартість, низьке енергоспоживання. 6loWPAN підтримує різні типи топологій, такі як сітчаста та

зіркова топологія. 6loWPAN пропонує адаптаційний рівень між рівнем MAC та мережевим рівнем (IPv6), щоб обробити сумісність між IEEE 802.15.4 та IPv6. Найбільш конкурентною альтернативою 6LoWPAN є ZigBee, як видно на рисунку 2.2. Обидва вони використовують один і той же протокол IEEE 802.15.4 на фізичному рівні, рисунок 2.2. приклади мереж. Стек протоколів 6LowsPAN та ZigBee [15]

Протокол ZigBee був створений Альянсом ZigBee на основі стандарту бездротових мереж IEEE802.15.4. ZigBee створений таким чином, що є стандартом для високоефективних протоколів зв'язку з низькою вартістю, створюючи персональні мережі з невеликих розмірів, цифрових радіостанцій малої потужності, які передають дані на більші відстані. в той же час він буде використовуватися в додатках, які вимагають низької швидкості передачі даних, більшого часу роботи від акумулятора та безпечних мережевих пристроїв. Крім того, ZigBee може підтримувати різні типи топологій, такі як сітка, зірка та деревоподібна топологія [15].

BLE також відомий як Bluetooth smart, що є важливим протоколом для програми IoT. Він розроблений та вдосконалений для короткого діапазону, низької пропускної здатності та низької затримки для додатків IoT. До переваг класичного Bluetooth BLE належать зменшення енергоспоживання, менший час налаштування та підтримка топології зоряної мережі з необмеженою кількістю вузлів [15].

RFID має різноманітні стандарти, включаючи (ISO, IEC, ASTM International, DASH7 Alliance та EPC-глобальні системи. RFID, що складаються з пристрою зчитування, який називається зчитувачем, і невеликого радіочастотного транспондера під назвою RF тег. Цей тег запрограмований в електронному вигляді з унікальною інформацією, яка має характеристику зчитування на відстані. Існує дві технології тегів RFID: перша називається системою тегів активних читачів, а друга називається пасивною системою тегів читачів. Вони дорогі та використовують більш високі частоти, тоді як пасивний тег використовує нижчі частоти і не має внутрішнього джерела живлення. Оскільки

інформація RFID є статичною і повинна бути запрограмована в тег, вона не може бути використана безпосередньо для вимірювань або діагностичних даних. додатків IoT, що використовують RFID, включають розумні покупки, охорону здоров’я, національну безпеку та сільське господарство. RFID може підтримувати топологію мережі P2P [16].

NFC – це технологія бездротового зв'язку дуже короткого діапазону, яка дозволяє передавати дані між пристроями, торкаючись їх разом або об'єднуючи їх не більше ніж на кілька дюймів. NFC використовує аналогічні принципи технології в RFID. Однак він використовується не тільки для ідентифікації, але й для більш досконалого двостороннього спілкування. NFC має тег, який може містити невелику кількість даних. Цей тег можна лише читати (подібно до тегів RFID для ідентифікації) або може бути перезаписаним та згодом змінено пристроєм. Технологія NFC широко використовується в мобільних телефонах, промислових додатках і безконтактних платіжних системах. Так само NFC полегшує підключення, введення в експлуатацію та управління IoT-пристроями в різних середовищах, таких як дома, на заводі та на роботі. NFC підтримує мережеву топологію P2P [17].

Z-Wave - це протокол MAC з низькою потужністю, розроблений Zensys, який використовує бездротову домашню автоматику для підключення 30-50 вузлів і використовується для IoT-зв'язку, особливо для розумного будинку та невеликих комерційних доменів. Ця технологія розроблена для невеликих пакетів даних з відносно низькою швидкістю до 100 кбіт / с і 30-метровим зв’язком. Тому він підходить для невеликих повідомлень у програмах IoT, таких як управління світлом, контроль енергії, контроль охорони здоров'я.

Z-Wave залежить від двох типів пристроїв (керуючих і ведених). Властивості ведених вузлів - це пристрої з низькою вартістю, які не можуть ініціювати повідомлення. Він може відповідати та виконувати лише команди, що надсилаються керуючими пристроями, які ініціюють повідомлення в мережі. Топологія сітки підтримки Z-Wave [18].

Рисунок 2.3. Покриття IoT IP. Адаптовано з роботи [18].

2.3 Порівняння протоколів комунікації

Розглянемо вибір правильного протоколу зв'язку шляхом надання порівняння між вищезазначеними протоколами зв'язку. Для порівняння відмінностей між протоколами зв'язку використовуються різні критерії. Такі критерії включають стандарт, мережу, топологію, потужність, діапазон, криптографію, розповсюдження, тип модуляції, механізм співіснування, безпеку та енергоспоживання, як показано в IoT покриті IP на рисунку 2.3 з точки зору безпеки, усі дев'ять протоколів зв'язку мають механізми шифрування та аутентифікації. 6LoWPAN, ZigBee, BLE, NFC, Z-Wave використовують блок-шифр розширеного стандарту шифрування (AES) у режимі лічильника, тоді як Cellular і RFID використовують RC4. Однак було виявлено кілька серйозних слабких місць. AES надзвичайно безпечний, а RC4 - ні. RC4 дуже швидкий порівняно з AES. З точки зору енергоспоживання, 6LoWPAN, ZigBee, BLE, ZWave та NFC розроблені для портативних пристроїв та обмеженої потужності акумулятора. Таким чином, він пропонує низьке енергоспоживання. З іншого

боку, в списку є висока енергоспоживання стільникового зв'язку. Щодо швидкості передачі даних, 6LoWPAN, ZigBee, BLE, NFC, SigFox та Z-Wave мають швидкість передачі даних <= 1 Мбіт / с. Однак RFID має найвищу швидкість передачі даних 4 Мбіт / с. У діапазоні SigFox та Cellular знаходяться в діапазоні довше, ніж покриття на кілька КМ. Однак 6LoWPAN, ZigBee, BLE, NFC, Z-Wave та RFID мають діапазон коротший, що покриває менше КМ. Відповідно до порівняння протоколу зв'язку в IoT, 6LoWPAN буде майбутнім протоколом, оскільки він має базується на IP-основі WSN. Це дозволяє легко розгорнути безліч розумних пристроїв через Інтернет, використовуючи величезний адресний простір IPv6 для збору даних та інформації за допомогою функцій та поведінки різних метрик, включаючи низьку пропускну здатність, різні топології та споживання енергії зіркою чи сіткою. , низька вартість, масштабовані мережі.

РОЗДІЛ 3
РОЗРОБКА ПРИНЦИПОВОЇ СХЕМИ ТА МІКРОКОДУ ДЛЯ КОНТРОЛЕРА ESP-8266

3.1 Розробка схеми приладу збору даних

У якості системи з віддаленим керуванням нами було розроблено інерційний блок вимірювання, це 3-осьовий акселерометр і 3-осьовий датчик гіроскопа. Акселерометр вимірює гравітаційне прискорення, а гіроскоп - швидкість обертання. Крім того, цей модуль також вимірює температуру. Цей датчик ідеально підходить для визначення орієнтації об’єкта, що рухається, також дозволяє передавати дані за каналом бездротового зв'язку на різні пристрої, наприклад планшети. Виконання поставленої задачі була здійснено у кілька етапів, першим із яких був проектуванням принципової електричної схеми експериментального приладу (рис. 3.2.). Внаслідок того, що нами використовувались комплектуючі, котрі виготовляються серійно, вона досить виявилась досить простою. У якості базового блоку (контролера) нами була обрана плата Node MCU з мікроконтролером ESP-8266, котра має у своєму складі розміщений контролер USB to TTL, який дозволяє виконувати прошивку мікрокодом без використання зовнішнього програматора, Wi-Fi чіп та стабілізатор напруги 3,3 В на базі мікросхеми ASM 1117 3.3. Використовуваний універсальний датчик MPU-6050 (рис. 3.1.)

Він використовує власний протокол зв'язку із контролером, а основні функції для роботи із ним описані в бібліотеках Adafruit_MPU6050 та Adafruit_Sensor, які і були використані нами.

MPU-6050 являє собою модуль з 3-осьовим акселерометром і 3-осьовим гіроскопом. Гіроскоп вимірює швидкість обертання (рад/с), це зміна кутового положення в часі вздовж осей X, Y і Z. Це дозволяє визначити орієнтацію об’єкта. Акселерометр вимірює прискорення (швидкість зміни швидкості руху об’єкта). Він відчуває статичні сили, такі як гравітація (9,8 м/с^2), або динамічні

сили, такі як вібрації або рух. MPU-6050 вимірює прискорення по осі X, Y та Z. В ідеалі в статичному об'єкті прискорення по осі Z дорівнює силі тяжіння, і воно повинно бути нулем на осях X і Y.

Використовуючи значення акселерометра, можна розрахувати кути нахилу та кроку за допомогою тригонометрії. Однак неможливо обчислити коливання.

Ми поєднали інформацію від обох датчиків, щоб отримати більш точну інформацію про орієнтацію датчика.

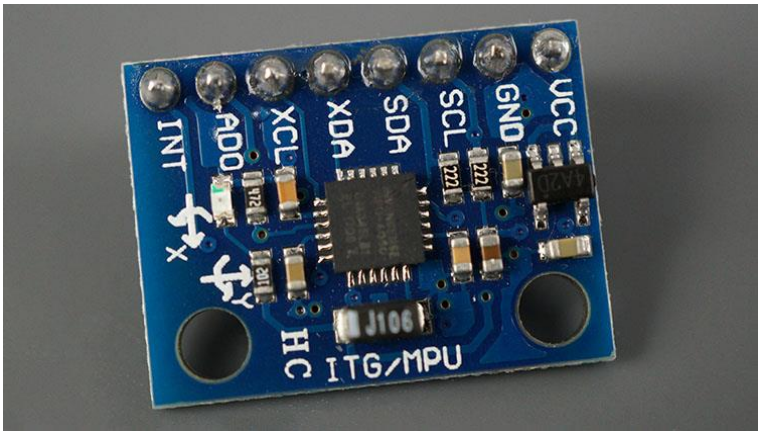


Рис. 3.1 – Зовнішній вигляд датчика MPU-6050.

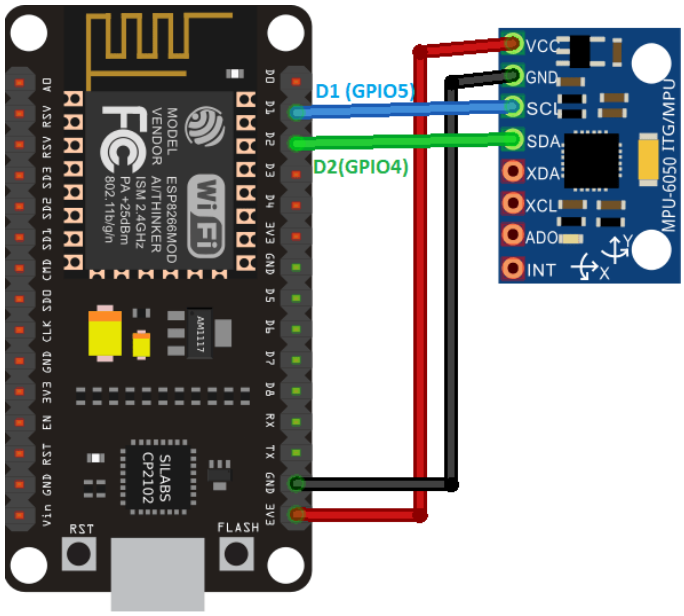


Рис. 3.2 – Схема інерційного блоку вимірювання на базі контролера ESP-8266 та датчика MPU-6050.

3.2 Програмне забезпечення контролера ESP-8266

Для контролера ESP-8266 нами було написане програмне забезпечення у середовищі програмування ArduinoIDE на мові програмування C++ із додаванням та підключенням спеціалізованих бібліотек для роботи з мікроконтролерами та датчиком MPU-6050. Першим функціональним блоком та етапом роботи програми є підключення зовнішніх бібліотек за допомогою директиви препроцесора #include. Частина коду для реалізації цих можливостей має наступний вид:

```
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
```

В наслідок виконання цього блоку нами був розширений стандартний функціонал програми для роботи з використовуваним нами обладнанням.

Наступним етапом роботи програми є ініціалізація датчика та налаштування діапазону його роботи, для цього нами була реалізована наступна частина коду:

```
if (!mpu.begin()) {
  Serial.println("Sensor init failed");
  while (1)
   yield();}
mpu.setAccelerometerRange(MPU6050_RANGE_8_G);
mpu.setGyroRange(MPU6050_RANGE_500_DEG);
mpu.setFilterBandwidth(MPU6050_BAND_5_HZ);
```

Далі в циклі loop() ми отримали показання датчиків і відобразили їх на послідовному моніторі. Для початку ми отримали нові події датчика з поточними показниками, отримуємо дані для прискорення (м / $с^2$) та дані з гіроскопа, кутова швидкість (рад / с) та температуру:

```
sensors_event_t a, g, temp;
mpu.getEvent(&a, &g, &temp);
Serial.print("Acceleration X: ");
Serial.print(a.acceleration.x);
```

```
Serial.print(", Y: ");
Serial.print(a.acceleration.y);
Serial.print(", Z: ");
Serial.print(a.acceleration.z);
Serial.println(" m/s^2");
Serial.print("Rotation X: ");
Serial.print(g.gyro.x);
Serial.print(", Y: ");
Serial.print(g.gyro.y);
Serial.print(", Z: ");
Serial.print(g.gyro.z);
Serial.println(" rad/s");
Serial.print("Temperature: ");
Serial.print(temp.temperature);
Serial.println(" degC");
  delay(500);  //Нові показання датчика відображаються кожні 500
```
мілісекунд.

Калібрування датчика відбувается в ідеалі, коли датчик статичний, значення гіроскопа повинні бути нульовими по всій осі, що не трапилось в нашому випадку. Коли датчик статичний, ми отримуємо такі значення гіроскопа: X: 0,06 рад/с Y: -0.02 рад/с Z: 0,00 рад/с. Тому потрібно враховувати помилку та виправляти значення в коді, щоб отримати точніші показники. Те саме відбувається зі значеннями прискорення. Прискорення по осі z повинно бути ближче до сили тяжіння (9,8 м/с^2) і повинно бути ближче до нуля на осі x та y. У нашому випадку це приблизні значення, які ми отримуємо, коли датчик статичний: X: 0,71 м/с^2 Y: 0,28 м/с^2 Z: 9,43 м/с^2.

Наступним кодом реалізовано відображення інформації на OLED – дисплей:

```
if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // 0x3C для 128x64
  Serial.println(F("SSD1306 allocation failed"));
```

```
  for (;;) ; } display.display();
display.setTextSize(1);
display.setTextColor(WHITE);
display.setRotation(0); loop()
display.clearDisplay();
display.setCursor(0, 0);
```

У наступних рядках відображаються показання акселерометра на послідовному моніторі:

```
Serial.print("Accelerometer ");
Serial.print("X: ");
Serial.print(a.acceleration.x, 1);
Serial.print(" m/s^2, ");
Serial.print("Y: ");
Serial.print(a.acceleration.y, 1);
Serial.print(" m/s^2, ");
Serial.print("Z: ");
Serial.print(a.acceleration.z, 1);
Serial.println(" m/s^2");
```

Наступні рядки відображають значення прискорення x, y a z на OLED-дисплеї:

```
display.println("Accelerometer - m/s^2");
display.print(a.acceleration.x, 1);
display.print(", ");
display.print(a.acceleration.y, 1);
display.print(", ");
display.print(a.acceleration.z, 1);
display.println("");
```

Відобразіть показання гіроскопа на послідовному моніторі.

```
Serial.print("Gyroscope ");
Serial.print("X: ");
```

```
Serial.print(g.gyro.x, 1);
Serial.print(" rps, ");
Serial.print("Y: ");
Serial.print(g.gyro.y, 1);
Serial.print(" rps, ");
Serial.print("Z: ");
Serial.print(g.gyro.z, 1);
Serial.println(" rps");
```

Виводимо данні гіроскопа на дисплеї.

```
display.println("Gyroscope - rps");
display.print(g.gyro.x, 1);
display.print(", ");
display.print(g.gyro.y, 1);
display.print(", ");
display.print(g.gyro.z, 1);
display.println("");
display.display(); delay(100);
```

У разі успішного виконання цього блоку коду програма виводить данні на дисплей(Рис 3.3).

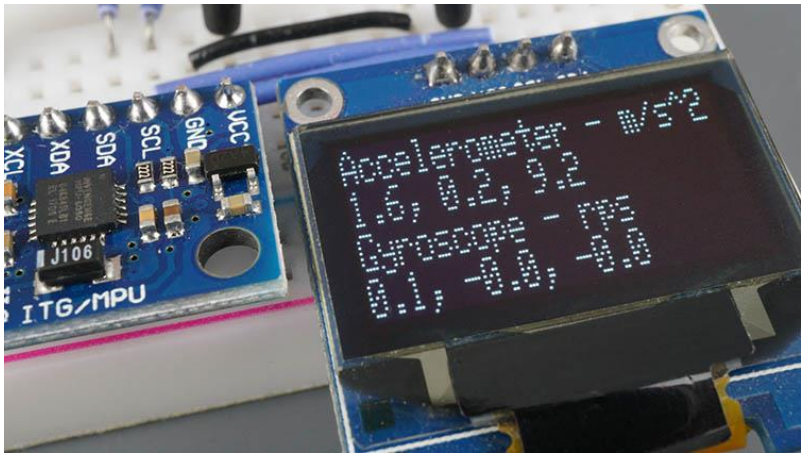


Рис. 3.3 – OLED-дісплей з виведеними даними.

Наступним блок коду це налаштування ESP8266 як точки доступу:

```
const char* ssid     = "ESP8266-Access-Point";
const char* password = "123456789";
.softAP(const char* ssid, const char* password, int channel, int ssid_hidden, int
max_connection)
IPAddress IP = WiFi.softAPIP();
Serial.print("AP IP address: ");
Serial.println(IP);
```

Результатом виконання котрого є налаштування комунікації із точкою доступу WI-FI.

Для забезпечення відповідності принципу IoT кожен окремий вузол повинен надавати свої інформаційні послуги користувачам і мати зручний комунікаційний інтерфейс. Одним з підходів, для забезпечення такого функціоналу, є розгортання Web-сервера безпосередньо на потужностях мікроконтролера із урахуванням його розрахункової потужності і апаратних засобів. Розгортання інформаційного публічного сервера на мікроконтролері ESP-8266 потребує доповнення коду написаної вище програми функціями реалізаціїї надання інформації за запитом а також задання формату і протоколу передачі даних.

Функціональний блок ініціалізації «void setup() {}», код якого виконується один раз при запуску контролера, необхідно доповнити операцією ініціалізацією з’єднання для передачі даних посередництвом послідовного інтерфейсу UART (Serial.begin(115200);). Також необхідно описати реалізацію підключення до мережі WI-FI та запуск Web-сервера на порту №80, дані операції реалізуються наступним функціональним блоком:

```
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
```

```
  Serial.print("."); }
 Serial.println("");
 Serial.println("WiFi connected");
 server.begin();
 Serial.println("Web server running. Waiting for the ESP IP...");
 delay(10000);
 Serial.println(WiFi.localIP()); }
```

У разі успішного виконання коду цього блоку буде забезпечено підключення до бездротової мережі, налаштовано інформаційний міні сервер трансляції даних, а також виведено повідомлення про це і службову інформацію щодо підключення (SSID та локальна IP адреса) до терміналу отримання даних за протоколом Serial.

Наступним блоком коду є цикл loop(), який виконується постійно, і потребує доповнення кодом обробки запитів. Логіка роботи даної частини коду наступна: на першому етапі програма буде чекати підключення клієнта до пристрою збору даних та надходження від нього запиту на отримання інформаційного пакету. Блок коду, котрий реалізує описані операції представлений таким чином:

```
WiFiClient client = server.available();
  if (client) {
    Serial.println("New client");
    // bolean to locate when the http request ends
    boolean blank_line = true;
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        if (c == '\n' && blank_line) {
```

Якщо операції, реалізовані у цьому блоці коду були завершені із статусом «успішно», то програмою буде здійснено перехід до перевірки зв’язку та правильного функціонування вимірювальних перетворювачів, і у разі успішного

його проходження, контролер почне зчитувати дані, використовуючи функції бібліотек та здійснить їх передачу до послідовного порту.

Головною ж задачею у цій роботі була побудова системи, яка б організувала передачу даних за допомогою закритого каналу безпровідного зв’язку. Дану задачу можна виконати багатьма різними способами, серед яких є написання окремого додатку для операційної системи комп’ютера, що виконуватиму функцію клієнтського програмного забезпечення, розробка мобільного додатку для IOS чи Android, або ж веб-сторінки. У рамках цієї роботи ми обрали останній варіант, оскільки від дає можливість керувати системою із використанням будь-якого пристрою, котрий підключений до мережі Internet, та одного із поширених веб-переглядачів.

Для керування віддаленою системою посередництвом сторінки в браузері, контролер потребує налаштування веб-сервера, для подальшого управління. Важливим моментом є те, що використовуючи виділений зовнішній сервер можна здійснювати доступ до системи користуючись доменним ім'ям. Забезпечити мінімальний функціонал веб-сторінки, використовуючи ресурси контролера Node-MCU, можна із застосуванням засобів мови гіпертекстової розмітки HTML. Застосовуючи такий підхід, щоправда, не можна використовувати елементи анімації, що у свою чергу унеможливлює створення насиченого веб-додатку, але функції відображення текстової інформації, за такого підходу, реалізуються досить добре. Створення такого міні сервера описувалось нами безпосередньо у мікрокоді контролера. Нами було написано код HTML сторінки для браузера, яка відображається на екрані користувача як відповідь на запит від клієнта:

```
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println("Connection: close");
client.println();
// your actual web page that displays temperature and humidity
client.println("<!DOCTYPE HTML>");
```

```
            client.println("<html>");
            client.println("<head></head><body><h1>ESP8266 – Axelerometer and Gyroscope</h1><h3> Accelerometer: ");
            client.println ("X: ");
            client.println (a.acceleration.x, 1);
            client.println (" m/s^2, ");
            client.println ("Y: ");
            client.println (a.acceleration.y, 1);
            client.println (" m/s^2, ");
            client.println ("Z: ");
            client.println (a.acceleration.z, 1);
            client.println (" m/s^2");
      client.println ("Gyroscope ");
      client.println ("X: ");
      client.println (g.gyro.x, 1);
      client.println (" rps, ");
      client.println ("Y: ");
      client.println (g.gyro.y, 1);
      client.println (" rps, ");
      client.println ("Z: ");
      client.println (g.gyro.z, 1);
      client.println (" rps");
            client.println("%</h3><h3>");
            client.println("</body></html>"); break; }
          if (c == '\n') {
            // when starts reading a new line
            blank_line = true; }
          else if (c != '\r') {
            // when finds a character on the current line
            blank_line = false;
```

Завершивши сеанс передавання даних до пристрою клієнта для відтворення у переглядачі програмою буде виконано останній етап своєї роботи, а саме відключення клієнта від каналу передачі інформації та закриття сесії передачі даних. Це виконується посередництвом наступних функцій:

```
client.stop();
Serial.println("Client disconnected.");
```

Отже, у рамках виконання завдання із побудови інерційного вимірювального пристрою став блок, реалізований на основі мікроконтролера ESP-8266, котрий може підключається точки доступу WI-FI з підключенням до мережі Internet, і має можливість передавати дані за запитом будь-якого пристрою, на котрому встановлено веб-переглядач. На рисунку 3.2 представлено результат відображення даних у браузері, отриманих вимірювальними перетворювачами.

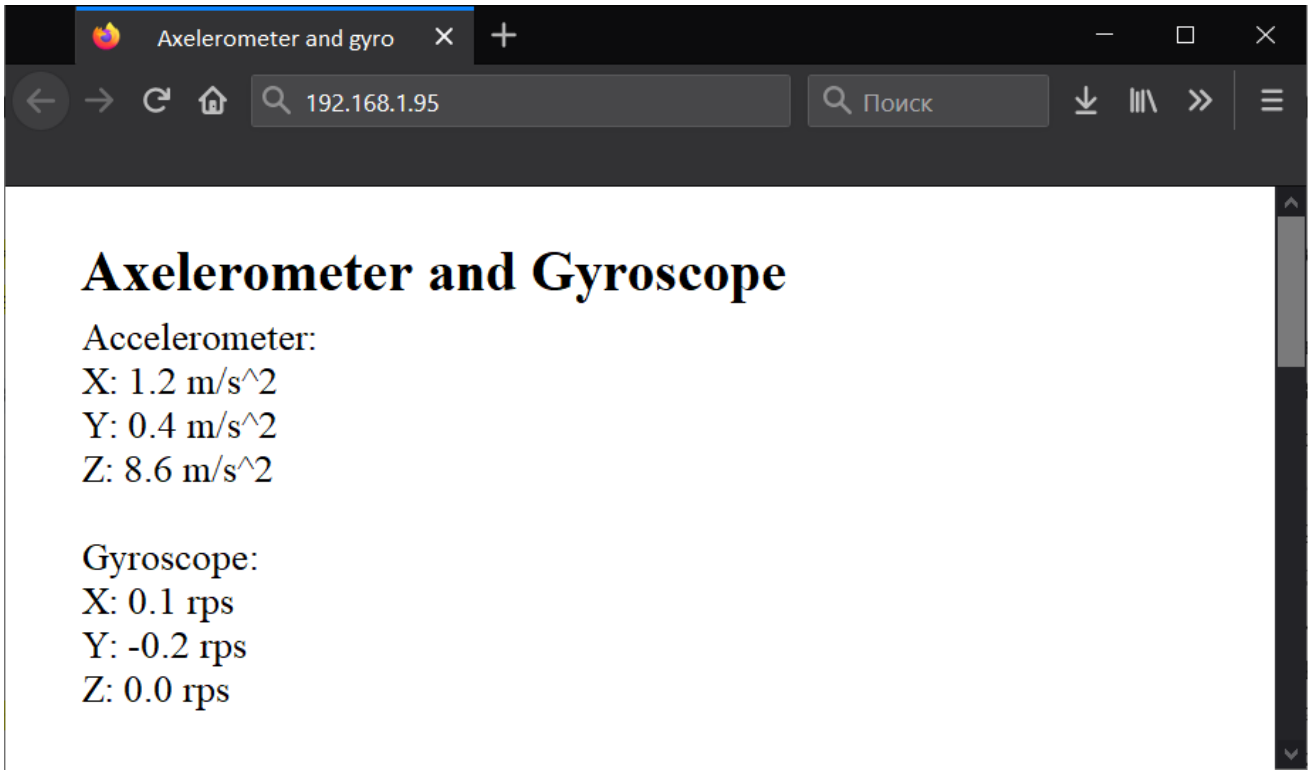


Рис. 3.2 – Сторінка браузера з отриманим даними

Використання принципів, встановлених у розробленому пристрої, дозволяє впровадити автоматизовану систему з можливістю дистанційного управління фізичним досвідом, а її функціональність забезпечується периферійними пристроями, підключеними до мікроконтролера. Використовуючи зовнішній веб-сервер як проміжне з'єднання між пристроєм і клієнтом, ви можете розробляти комплексні веб-програми з розширеними функціями та зручним інтерфейсом. Не складе труднощів реалізувати анімацію та відстеження, які візуально візуалізують фізичні процеси, що контролюються контролером та приєднаними до нього датчиками.

ВИСНОВКИ

1. Література мікроконтролерів AVR та ESP була ретельно розглянута та проаналізована. Здійснено огляд моделей вимірювальних платформ масового виробництва.

2. Враховуються технічні властивості та властивості мікроконтролерів сімейств AVR та ESP та апаратних платформ на їх основі, що випускаються серійно. Розглянуто переваги та недоліки окремих моделей.

3. Апаратні платформи, засновані на процесорах AVR та ESP, виявилися універсальними елементами управління, які можна запрограмувати на конкретне завдання і стати повноцінним електронним пристроєм для будь-яких цілей. Ці мікроконтролери запрограмовані на мові C ++, що простіше, ніж низько рівневі мови ассемблера, які раніше використовувались для програмування мікроконтролерів.

4. Розглянуто програмування модулів Wi-Fi на базі мікроконтролера ESP-8266 в середовищі програмування Arduino IDE. Показано, що система моніторингу та контролю для фізичного експерименту може керуватись через Інтернет на основі цього контролю.

5. За допомогою побудованого інерціального вимірювального блоку на основі контролера ESP8266 ми можемо контролювати та отримувати інформацію на пристрої які маюсьдоступ до інтернету.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ambika K., Malliga S. Epitome Evolution of Sanctuary to Detect the Interloper in Home Automation. – Erode : Proceedings of ICTIDS, 2020. – 629 p.

2. Barrett S. F. Arduino I: Getting Started. – Laramie : University of Wyoming, 2020. – 222 p.

3. Behara B. Mhetre M. Real Time Health Monitoring System Using IoT. – Pune : Springer, 2020. – 132 p

4. Betty Jane J., Ganesh E. N. Big Data and Internet of Things for Smart Data Analytics Using Machine Learning Techniques. – Chennai : Springer Science and Business Media Deutschland GmbH, 2020. – 223 p

5. Chandrasekaran S., Veeran R. Assistive device for neurodegenerative disease patients using iot. – Bangalore : Springer Science and Business Media Deutschland GmbH, 2020. – 452 p.

6. Chauhan R., Punj D., Joshi R. C. RFID-Based Smart Shopping in IoT Environment: A Way to Become a Smart Shopper. – Dehradun : Springer, 2020. – 380 p.

7. Diván M. J., Sánchez Reynoso M. L. An architecture for the real-time data stream monitoring in IoT. – Santa Rosa : Springer Science and Business Media Deutschland GmbH, 2020. – 100 p.

8. Gollmer K. IoT-workshop: Blueprint for pupils education in IoT. –Trier : Elsevier, 2019. – 344 p.

9. Gowri S. Implementation of IOT in Multiple Functions Robotic Arm: A Survey. – Chennai : Springer Science and Business Media Deutschland GmbH. 2020. – 952 p.

10. Goyal L. M., Mittal M., Sharma A. Automation movie recommender system based on internet of things and clustering. – Wiley : Scrivener Publishing LLC, 2019. – 128 p.

11. Guchhait P., Sehgal P., Aski V. J. Sensoponics: IoT-Enabled Automated Smart Irrigation and Soil Composition Monitoring System. – Jaipur : Springer Verlag, 2020. – 101 p.

12. Hassnuddin M., Kumar S., Dalmia H. Electricity Management in Smart Grid Using IoT. – Hyderabad : Springer, 2020. – 337 p.

13. Joshi V. B., Goudar R. H. IoT-based automated solution to irrigation: An approach to control electric motors through Android phones. – Belagavi : Springer Verlag, 2019. – 330 p.

14. Kumari Shibani K. S. S. K. and G. S. S. Artificial Intelligence Engineering Systems Computations in and Evolutionary. – New Delhi : Springer India, 2020. – 77p.

15. Kurniawan A. Arduino Nano 33 IoT Networking. – Berkeley : Apress, 2021. – 161 p.

16. Kurniawan A. IoT Projects with Arduino Nano 33 BLE Sense. –Berkeley : Apress, 2021. – 129 p.

17. Mabrouki J. Intelligent System for Monitoring and Detecting Water Quality Springer. – Rabat : Springer, 2020. –182 p.

18. Martín-Garín A. IoT and cloud computing for building energy efficiency. – Burgos : Elsevier, 2020. – 265 p.

19. Mori G. N., Swaminarayan P. R. Measuring IoT Security Issues and Control Home Lighting System by Android Application Using Arduino Uno and HC-05 Bluetooth Module. – Vadodara : Springer Science and Business Media Deutschland GmbH, 2021. – 382p.

20. Pachipala Y. Interactive Video Gaming with Internet of Things. – Guntur : Springer Science and Business Media Deutschland GmbH, 2020. – 445 p.

21. Тищенко К.В., Кучменко В.Г., Однодворець К.С. Розподілені мережі в системах моніторингу та малої автоматизації / Матеріали Міжнародної науково-технічної конференції студентів та молодих вчених «Фізика, електроніка, електротехніка ФЕЕ-2021». – Суми: СумДУ, 2021. – С.61.